



Universitat de Lleida

TREBALL FINAL DE GRAU



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: Lluís Oromí Cuello

Titulació: Grau en Enginyeria Informàtica

Títol de Treball Final de Grau: Galactic Conquest: Un joc d'estratègia 4X fet amb Unity

Director/a: Francesc Sebé Feixas

Presentació

Mes: Setembre

Any: 2018

Index

Objectius	4
Introducció al joc.....	6
Generació del mapa galàctic	7
Investigació.....	10
Àrees d'investigació.....	11
Control de colònies.	13
Disseny de naus.....	15
Bombardeig planetari.....	17
Escenes.....	21
Classes	23
Classes "MonoBehaviour"	23
Classe "GameController"	23
Classe "BombingController"	24
Classe "TechManager"	25
Classe "ShipyardsManager"	26
Classes objecte	27
Classe "Solar System"	27
Classe "Planet"	28
Aspectes a afegir o millorar en futures versions del joc	30
Annex	31
Tipus d'estrelles i les seves característiques.....	31
Tipus de climes dels planetes.....	34
Conclusions	37
Bibliografia	38

Index de figures

Figura 1 Galàxia en forma d'anell amb 10 sistemes solars i 4 jugadors.....	7
Figura 2 Galàxia en forma de disc amb 10 sistemes solars i 2 jugadors	8
Figura 3 Pantalla d'informació d'un sistema solar amb informació detallada d'una estrella	8
Figura 4 Pantalla d'informació d'un sistema solar amb informació detallada d'un planeta.....	9
Figura 5 Pantalla d'investigació.....	10
Figura 6 Pantalla de control de colònia d'un planeta de tipus terraqui.....	13
Figura 7 Panell de construcció	14
Figura 8 Pantalla de disseny de naus	15
Figura 9 Pantalla de bombardeig planetari.....	17
Figura 10 Bombardejant un planeta desèrtic.....	19
Figura 11 Bombardejant un planeta àrid	19
Figura 12 Bombardeig en curs.....	20

Objectius

L'objectiu d'aquest TFG es desenvolupar un joc d'estratègia espacial del tipus 4X per tornos en Unity.

El terme 4X prové d'un anàlisi que es va fer del joc "Master of Orion" l'any 1993. És una abreviació de les quatre fases en les que es pot dividir el joc, són "eXplorar, eXpandir, eXplotar i eXterminar".

- *Explorar* acostuma a ser al principi del joc i els jugadors envien exploradors per tot el mapa per tal de trobar els millors recursos.
- *Expandir* té lloc quan els jugadors reclamen territori, ja sigui per estendre la seva influència en el mapa o per accedir amb més facilitat als recursos.
- *Exploitar* fa referència a recollir i utilitzar aquests recursos.
- *Exterminar* normalment es refereix a eliminar els altres jugadors, tot i que també hi ha jocs en que és necessari eliminar obstacles per a poder avançar.

Tot i que aquestes són les quatre fases que tenen lloc en un joc 4X normal i corrent, normalment aquestes es solapen i unes duren més que les altres. Per exemple, els jocs que fan servir mapes molt grans acostumen a tenir una fase d'exploració que dura pràcticament tota la partida.

Els jocs d'estratègia 4X es solen diferenciar de la resta de jocs d'estratègia per la seva complexitat i duració.

Acostumen a tenir un arbre tecnològic i al contrari que en els jocs d'estratègia normals, per al progrés tecnològic no es necessita que es construeixin edificis o unitats especials per a desbloquejar tecnologies. A més a més, la tecnologia sol tenir un pes molt important en els combats, fins al punt que una partida es pot donar per guanyada o per perduda depenent de la diferència de tecnologies entre els jugadors.

Un altre factor important que els diferencia dels altres tipus de jocs d'estratègia és la presència de diplomàcia. Depenent del joc inclús és possible guanyar una partida sense atacar ningú.

Exemples de jocs 4X:

- La saga *Civilization*: són 4X per torns ambientats a la Terra
- La saga *Sword of the Stars*: són 4X per torns, però amb combat en temps real, ambientats a l'espai.
- La saga *Age of Wonders*: són 4X per torns ambientats en un món de fantasia.
- *Sins of a solar Empire* és un 4X en temps real ambientat a l'espai.
- La saga *Masters of orion*: són 4X per torns ambientats a l'espai.

Sempre m'han agradat els jocs d'estratègia, i des de que vaig provar el "Masters of Orion 2" he anat jugant a la major part de jocs 4X que han sortit al mercat.

A l'hora de fer aquest joc, he pres com a referent el "Masters of Orion 2", ja que a més a més de ser el primer joc 4X al que vaig jugar, segueix sent considerat el millor joc del seu gènere, tot i que ja fa més de 20 anys que va sortir al mercat.

Introducció al joc

Galactic Conquest és un joc d'estratègia 4X per torns ambientat a l'espai.

En cada partida, el jugador es fa càrrec d'una civilització i ha d'assegurar-ne la seva supervivència eliminant la resta de jugadors.

Per tal de facilitar la feina d'eliminar els rivals, el joc disposa d'un sistema d'investigació de tecnologies que desbloquejen nous edificis i components per a les naus.

Aquests edificis es poden construir en els planetes per tal d'obtenir un avantatge sobre els rivals, i els components de les naus serveixen per a fabricar naus que poden servir tant per a atacar o per a defensar planetes.

La partida s'acaba quan només queda un jugador en tot el mapa. Un jugador és eliminat quan es queda sense planetes. Es poden perdre planetes degut a un bombardeig, a una invasió terrestre o si tota la població del planeta es mor de gana.

Generació del mapa galàctic

Un cop s'han seleccionat els paràmetres de la partida es procedeix a la generació del mapa galàctic.

En primer lloc es té en compte quina forma té la galàxia. Si té forma d'anell les coordenades X i Y de cada sistema solar venen donades per les fórmules següents:

$$\text{Posició X} = \cos(360 / \text{nombre total d'estrelles}) * \text{índex},$$

$$\text{Posició Y} = \sin(360 / \text{nombre total d'estrelles}) * \text{índex},$$

On índex és el nombre del sistema solar que s'està creant en aquell moment, i va des de 0 per al primer sistema solar fins a (nombre total d'estrelles) - 1.



Figura 1 Galàxia en forma d'anell amb 10 sistemes solars i 4 jugadors

En canvi, si la galàxia té forma de disc, les coordenades x i y es generen aleatòriament.



Figura 2 Galàxia en forma de disc amb 10 sistemes solars i 2 jugadors

Cada cop que es crea un sistema solar, aquest escull un tipus d'estrella aleatòriament. El tipus de l'estrella determinarà quins climes tindran els planetes del sistema solar. Cada sistema pot tenir entre 1 i 5 planetes.

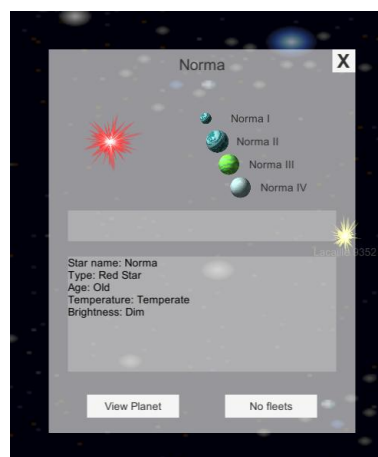


Figura 3 Pantalla d'informació d'un sistema solar amb informació detallada d'una estrella

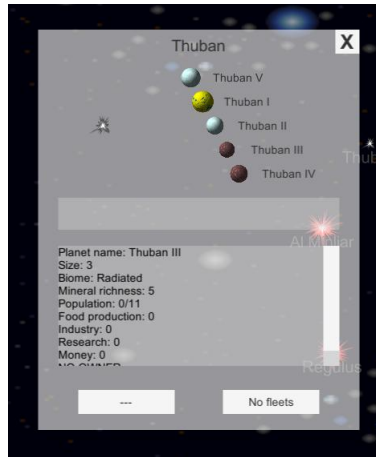


Figura 4 Pantalla d'informació d'un sistema solar amb informació detallada d'un planeta

Un planeta té una mida, que influeix en el nombre màxim de població que pot tenir, i una riquesa mineral, que proporciona un bonus a la construcció de noves naus i edificis. Addicionalment, cada planeta té un clima el qual determina la facilitat amb la que es pot alimentar la població existent. Els planetes amb climes més durs requereixen edificis que produeixin menjar o que la major part de la població es dediqui a cultivar per a poder sobreviure.

Un cop s'han acabat de generar tots els sistemes solars amb els seus respectius planetes es passa a assignar un planeta a cada jugador. Per tal de jugar en igualtat de condicions, al assignar un planeta a un jugador, aquest passa a tenir mida i riquesa mineral normals, i es canvia el clima a "Terraquí". També s'incrementa la població a 5 unitats i s'assignen de forma automàtica per a que la població sigui autosuficient i no morin d'inanició.

Investigació

Un dels elements més importants del joc és la investigació de noves tecnologies. Un jugador amb un nivell tecnològic elevat podrà construir i investigar més ràpid que un altre jugador amb un nivell tecnològic més baix. Un petit avantatge tecnològic al principi de la partida s'acostuma a accentuar al llarg de la partida i pot acabar decidint qui és el guanyador.

L'arbre tecnològic és molt simple. Consisteix en 7 àrees d'investigació sense interdependències entre elles. És possible investigar una única àrea fins al final i ignorar la resta, tot i que no és una bona estratègia.

Cada àrea d'investigació està formada per una seqüència de nivells, i cada un d'ells desbloqueja un o més edificis o components per a les naus.

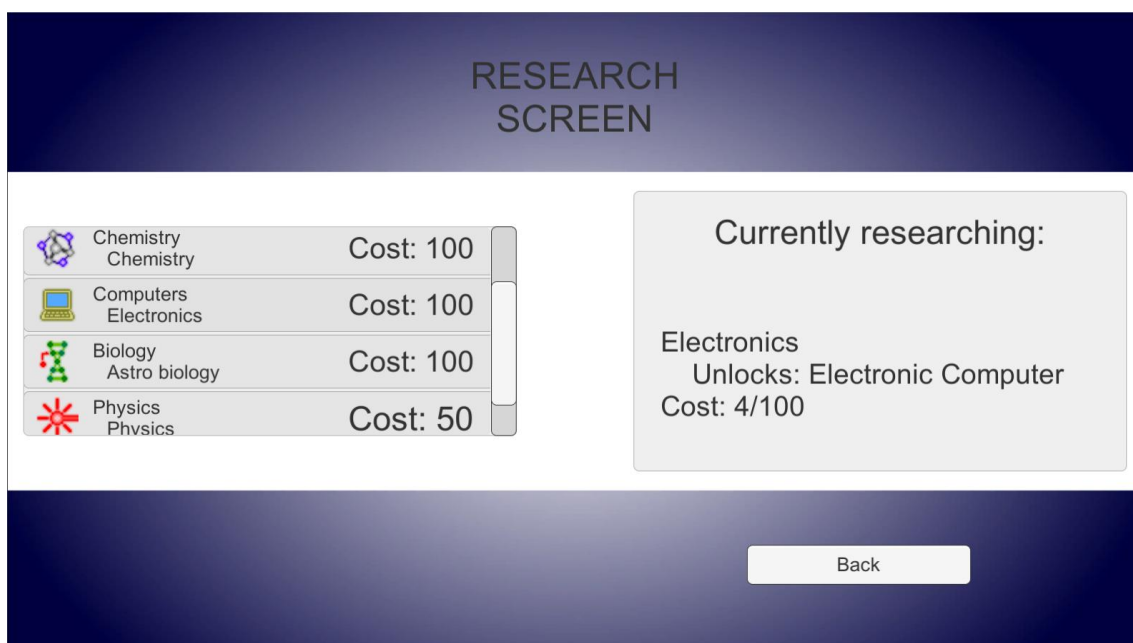
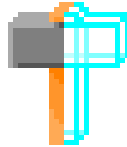


Figura 5 Pantalla d'investigació

Àrees d'investigació

Totes les tecnologies de l'arbre tecnològic es poden agrupar en set àrees d'investigació diferents, cada una d'elles enfocada en diferents aspectes de l'imperi del jugador. Les àrees d'investigació són les següents:

- Enginyeria: Enfocada a desenvolupar nous edificis i desenvolupar noves formes d'incrementar la producció.



- Energia: Enfocada a desenvolupar noves fonts d'energia i les seves aplicacions.



- Química: Enfocada a dissenyar nous materials i les seves aplicacions.



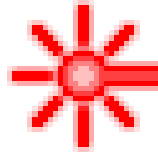
- Electrònica: Encarada a desenvolupar noves formes de processar informació, ja sigui incrementant la producció de recerca o fent que les armes de les naus siguin més precises.



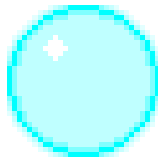
- Biologia: S'encarrega d'investigar noves formes de vida i com fer-les servir per poder alimentar la població.



- Física: S'encarrega del disseny i la manufacturació de la major part de les armes del joc.



- Camps energètics: S'encarrega d'investigar i desenvolupar noves tecnologies basades en l'aplicació de camps energètics. El seu ús principal és el d'investigar escuts per a les naus.



Cada nivell de cada branca té un cost en punts de recerca que es va incrementant amb cada nova tecnologia que s'investiga.

Al final de cada torn, es sumen tots els punts de recerca generats a tots els planetes del jugador i es sumen als punts acumulats. Si el resultat és més gran o igual que el cost de la nova tecnologia, aquesta es desbloqueja.

L'implementació de l'arbre tecnològic es fa mitjançant un fitxer XML que el conté. Cada tecnologia de l'arbre es un camp "tech" i disposa d'un identificador, que en aquest cas és el nom.

A més a més disposen de camps addicionals, uns que són comuns a totes les tecnologies, com per exemple a quin camp tecnològic pertanyen i quin cost d'investigació tenen, i d'altres que poden variar entre tecnologies.

L'avantatge de fer servir un fitxer XML per emmagatzemar les dades de l'arbre tecnològic enlloc de tindre les dades "hardcodejades" dintre d'alguna altra classe és que permet afegir, treure i manipular les tecnologies amb molta facilitat, ja que només es requereixen coneixements bàsics per tractar fitxers XML.

Per aquest mateix motiu la llista de components de les naus i la llista d'edificis també es troben emmagatzemades en fitxers XML

Control de colònies.

Un cop s'ha colonitzat un planeta, al fer clic sobre la seva imatge a la pantalla d'informació del sistema solar i fer clic sobre el botó "View Planet", s'obrirà una nova pantalla amb les dades pertinents sobre el planeta seleccionat.

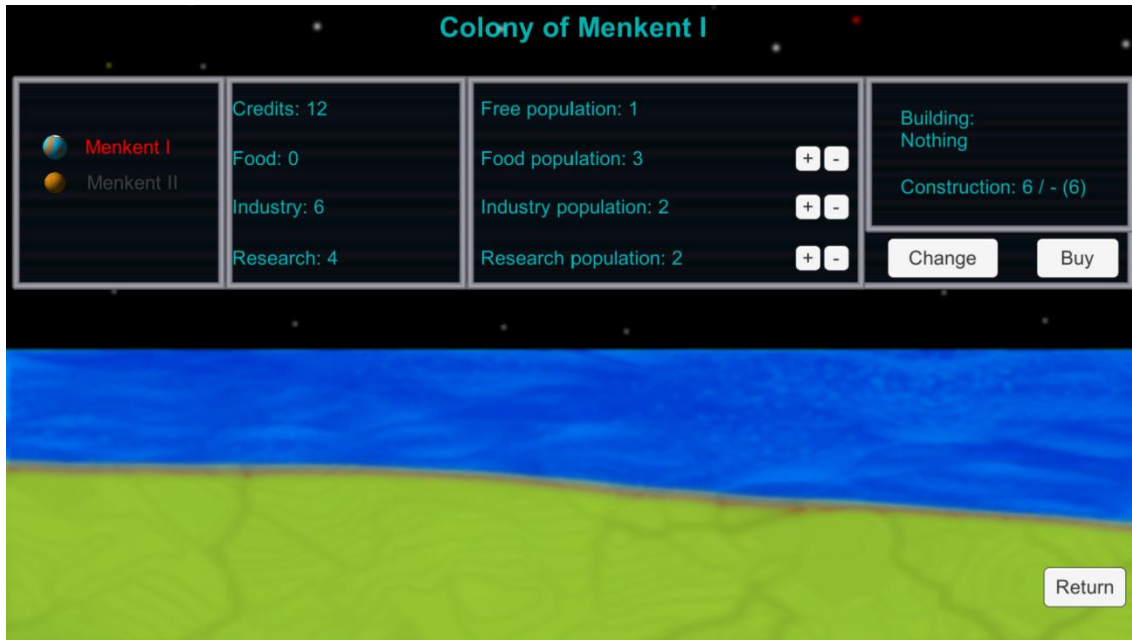


Figura 6 Pantalla de control de colònia d'un planeta de tipus terraqui

A la part superior de la pantalla hi ha un panell informatiu amb tota l'informació important del planeta:

- A l'esquerra de tot es troba un panell amb una representació en miniatura del sistema solar amb els seus planetes.
- Més a la dreta es troba el panell de producció, on es pot veure quants diners, menjar, producció i investigació s'estan produint actualment.
- Una mica més a la dreta es troba el panell de control de població. En aquest panell es pot veure quanta gent treballa produint menjar, fabricant naus o edificis, o investigant noves tecnologies. A més a més es pot assignar i treure gent mitjançant els botons.
- A la dreta de tot es troba el panell de construcció. Aquí es pot veure si s'està construint un edifici o una nau i quin progrés porta.

Al panell de construcció hi ha un botó que obre la finestra de construcció.

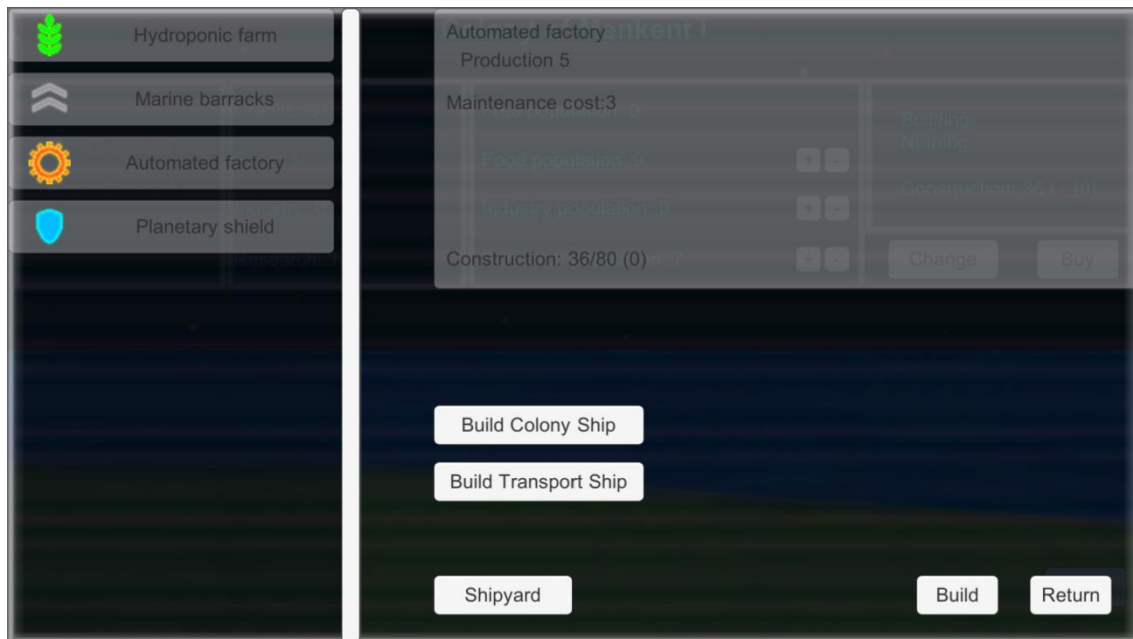


Figura 7 Panell de construcció

En aquesta pantalla podem seleccionar, en llista, quin edifici volem construir d'entre tots els edificis que hem desbloquejat mitjançant la recerca. A més a més també podem construir naus de colonització i d'invasió.

Per últim, destacar el botó de dissenyar nau ("Shipyard"). Al fer clic en aquest botó s'obrirà la pantalla de dissenyar naus.

Disseny de naus

Al fer clic al botó de "Shipyard" al panell de construcció a la pantalla de control de colònies, s'obre la pantalla de disseny de naus.

En aquest joc les naus estan formades per components, i aquests es poden dividir en tres categories:

- Components base: Es seleccionen automàticament a l'hora de construir una nau, i sempre s'agafa la millor versió de cada categoria de components. Les categories són: motors, blindatge, escuts i ordinadors. D'aquestes quatre categories les dos primeres són requisits indispensables a l'hora de fabricar naus.
- Components ofensius: Com el seu nom indica, aquests components són els que permeten a les naus atacar i defensar-se. D'aquest grup cal destacar les bombes, ja que són aquestes les que permeten fer bombardeigs planetaris.
- Components especials: Proporcionen bonificacions especials a la nau i al contrari que els components ofensius, no hi poden haver components repetits.

Cada nau té una mida i aquesta determina quants components hi caben. A més a més, la vida i el cost d'una nau venen determinat per la mida.

My first ship

Ship designer

Ship class
Battleship

Phase Drive
6 parsecs/turn
30 combat speed

Xentronium Armor
128 structure points
128 armor points

Class X shield
100 strength
10 damage points blocked
Moleculartronic Computer
125 beam attack

Name	Damage	Range	Cost	Space	Count
Graviton Beam	5	60	80	25	+ 4 -
Disruptor	100	20	150	50	+ 2 -
Reinforced hull		300			
Heavy armor		400			

Add Weapon Remove Weapon

Add System Remove System

Cost 2390 Space Available 15

Clear Cancel Build

Figura 8 Pantalla de disseny de naus

La pantalla de disseny de naus està formada per un menú amb múltiples parts:

- A dalt de tot tenim un camp de text on posar el nom que li volem donar a la nau. Si es deixa en blanc, el joc li assigna un nom per defecte.
- Al centre de la pantalla tenim un desplegable on podem seleccionar la mida de la nau. Al canviar la mida de la nau s'esborrarà qualsevol component que ja s'hagi afegit a la nau, així que és preferible que sigui el primer camp a seleccionar.
- A la part superior dreta tenim un panell amb informació sobre els quatre components base que te instal·lats la nau.
- A la part central tenim les llistes de components ofensius i components especials, juntament amb els botons que permeten afegir i treure'ls. Al fer clic al botó d'afegir un component apareixerà una llista amb tots els components d'aquell tipus disponibles i al seleccionar-ne un s'afegirà a la llista de components instal·lats. Si es fa clic sobre un component instal·lat i després es clica al botó de treure components llavors aquest s'elimina.
- A la part inferior tenim els comptadors del cost de la nau i de l'espai disponible. El cost de la nau es calcula en punts de producció i mentre hi hagi espai disponible es podran anar afegint més components. No es podran afegir components si aquests ocupen més que l'espai disponible, o si ja hi ha massa components.
- A baix a la dreta hi ha els botons d'esborrar, de confirmar la construcció o de cancel·lar-la i tornar a la pantalla anterior. Es pot confirmar la construcció de qualsevol tipus de nau sempre i quan l'espai disponible sigui igual o més gran que zero.

Bombardeig planetari

L'objectiu del bombardeig planetari es eliminar tropes enemigues per a poder conquerir el planeta de l'enemic.



Figura 9 Pantalla de bombardeig planetari

A la part superior de la pantalla es pot veure el nom del planeta enemic, i les icones dels jugadors enfrontats, a l'esquerra l'atacant i a la dreta el defensor.

A la part dreta de la pantalla es troben les dades del planeta del defensor: nombre d'habitants, nombre d'edificis i el nombre de tropes terrestres i tancs que té el planeta per a defensar-se d'una invasió.

Adicionalment, també hi apareix un comptador de la pol·lució que té el planeta i un altre indicador que diu quanta vida té l'escut planetari en cas d'haver-ne un construït.

A la part esquerra de la pantalla es troben els comptadors de bombes de la flota atacant, ordenats per tipus: bombes convencionals, de destrucció massiva, biològiques i químiques, juntament amb els botons per a deixar-les caure sobre el planeta. A sota de tot es troba el comptador de tropes terrestres i el botó per a envair el planeta.

Al centre de la pantalla es pot veure una representació del planeta atacat, a on es pot veure si el planeta té escut planetari o no, i també es poden veure les explosions de les bombes.

El funcionament dels bombardeigs es molt senzill:

Si el planeta té un escut planetari, aquest farà que qualsevol bomba que no sigui de destrucció massiva no faci danys, a part d'impedir una invasió terrestre. Un cop l'escut planetari ha caigut és quan els diversos tipus de bombes entren en joc:

Les bombes convencionals tenen molt poques probabilitats de matar població o destruir edificis, però són barates i ocupen molt poc espai, de manera que pel cost d'una bomba de destrucció massiva pots tenir múltiples bombes convencionals.

Les armes de destrucció massiva són les úniques que poden destruir escuts, i a més a més són bastant efectives a l'hora d'eliminar població i edificis, però tenen l'inconvenient de que són cares i ocupen molt espai.

Les armes biològiques tenen l'avantatge de que no destrueixen edificis ja que només afecten a la població i a les tropes enemigues, facilitant una posterior invasió. Però comporten el risc de que al final s'acabi envaint un planeta sense població, de manera que el jugador acabaria perdent-lo.

Les armes químiques serveixen per empitjorar les condicions climàtiques d'un planeta. Quan un planeta arriba al 100% de pol·lució, canvia el seu clima, empitjorant la vida dels seus habitants i fent que sigui més difícil que hi hagi una població. En el pitjor dels casos, el planeta queda tant contaminat que és impossible viure-hi.

L'objectiu de les bombes és reduir el nombre de tropes terrestres enemigues per a que les tropes atacants puguin eliminar-les i capturar el planeta. El combat terrestre segueix la fórmula matemàtica següent: $\text{força_atacants} - (\text{força_tropes_defensores} + \text{força_tancs_defensors} * 2)$. Si el resultat és positiu, el planeta passa a ser capturat i s'acaba la batalla. Si no, es resta a les forces defensores el valor de la força atacant. Per exemple, si envaeixo un planeta de 10 tropes amb un exèrcit de 4, perdré el combat, però el planeta acabarà amb 6 tropes.

A nivell d'Unity, la pantalla de bombardeig és un panell que serveix com a fons de pantalla, un panell a cada costat, cada un amb la seva informació pertinent dins d'un altre grup de panells, i dos sprites¹ al centre de la pantalla: un per l'imatge del planeta i un altre per l'escut planetari.

Quan s'obre la pantalla de bombardeig, la classe controladora de la pantalla de bombardeig recolleix tota l'informació necessària de la flota atacant (nombre de bombes i tropes terrestres) i del planeta atacat (població, nombre d'edificis, etc.) i actualitza els comptadors pertinents. A continuació canvia el sprite del planeta depenent del tipus de clima del planeta. A més a més, si el planeta disposa d'un escut, fa que sigui visible el sprite de l'escut planetari.

¹ Els sprites són objectes en dos dimensions utilitzats per a representar objectes en un entorn de joc bidimensional



Figura 10 Bombardejant un planeta desèrtic

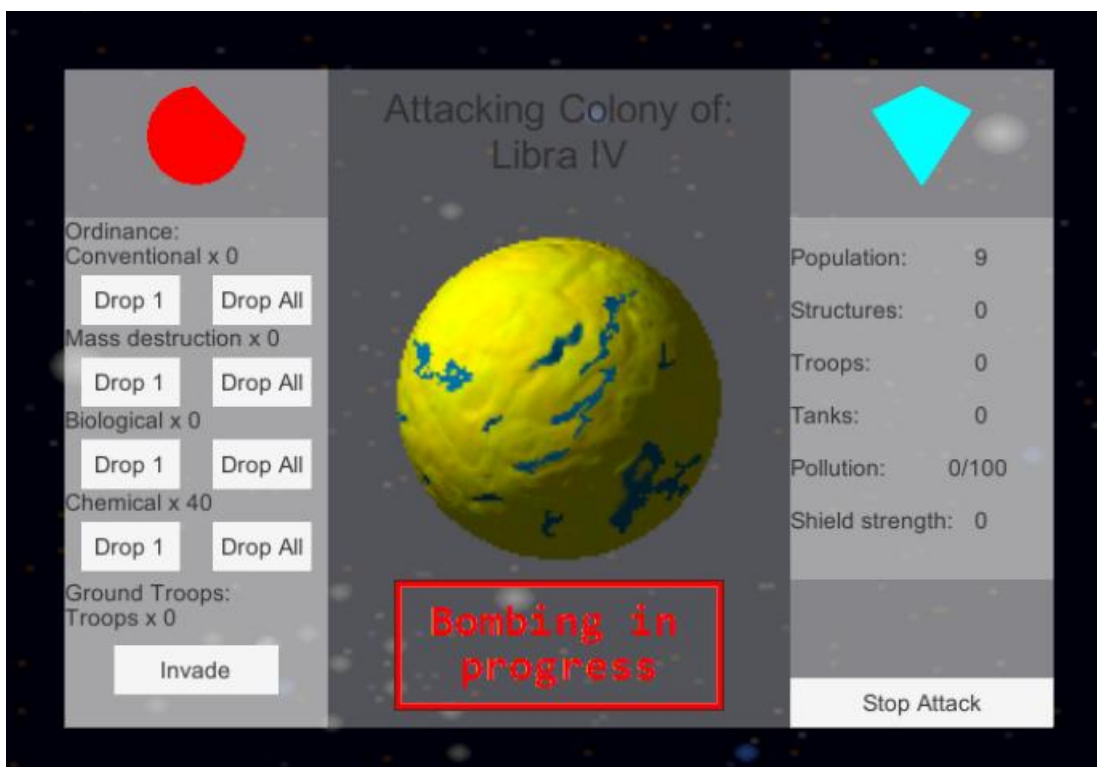


Figura 11 Bombardejant un planeta àrid

Per als efectes de les explosions de les bombes s'han creat quatre sprites mitjançant l'eina GIMP i han estat animats amb Unity. Cada cop que es tira una bomba al planeta, es crea un objecte del tipus sprite amb el sprite corresponent en una posició aleatòria de la imatge del planeta.

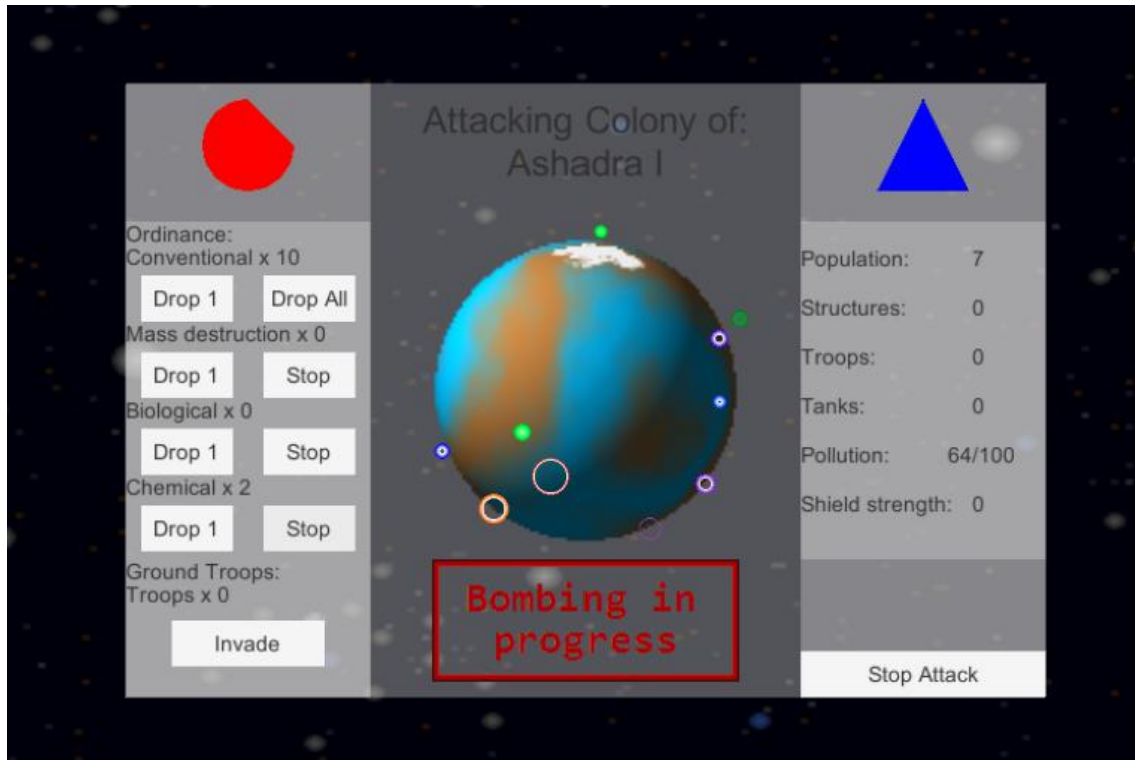


Figura 12 Bombardeig en curs

Escenes

Una de les característiques de la plataforma de desenvolupament Unity és l'ús d'escenes. Cada escena es pot considerar com un nivell del joc. Es poden fer servir per a fer des de pantalles de menús a nivells del joc.

En aquest joc hi ha cinc escenes diferents i una que no es fa servir. Les escenes són les següents:

- Menú principal. La idea d'aquesta escena era tenir una pantalla des d'on poder configurar els paràmetres de la partida, treballar amb desplegable i aprendre a transferir dades entre objectes.
- Pantalla d'investigació. En aquesta escena volia posar un menú en el qual poder escollir la següent tecnologia a investigar. Al final vaig fer una llista de prefabs² on cadascun és una tecnologia. Al seleccionar-ne un apareix informació detallada. Gràcies a aquesta escena he pogut desenvolupar un mètode per a mostrar i seleccionar elements d'una llista creada a partir d'un fitxer XML, el qual he fet servir en la resta d'escenes.
- Pantalla de construcció de naus. Aquesta escena consta d'un únic menú amb desplegable, botons i dos llistes de prefabs. En aquesta escena he pogut desenvolupar una mica més els prefabs, fer-los dinàmics i poder crear un objecte (una nau en aquest cas) amb les dades que vol l'usuari i passar aquestes dades d'una escena a una altra sense perdre informació.
- Pantalla de control de colònia. Aquesta escena conté un menú format per múltiples panells i imatges de fons. Aquí he pogut provar mètodes per a canviar les imatges de fons depenent de certes condicions, i canviar dades de forma dinàmica d'un objecte. Addicionalment m'ha servit per aprendre a crear pantalles superposades i poder-les manipular durant la partida sense que hi hagi interferències amb els objectes de darrere.
- Mapa galàctic. És l'escena principal del joc i en ella he aplicat tots els conceptes que he anat aprenent al llarg del projecte. Des de finestres emergents com la vista d'informació d'un sistema solar a llistes de prefabs en el cas de la vista de flota. També he après a crear objectes que puguin ser clicats sense necessitat de que siguin botons, o fins i tot simular canvis d'escena amb el bombardeig planetari, gràcies al qual vaig aprendre a utilitzar recursos d'àudio.

² Un prefab és una plantilla d'un objecte. D'aquesta manera es poden guardar els seus paràmetres i no cal canviar-los cada cop que es crea una nova instància d'aquest objecte.

L'escena que no es fa servir és la que pertany al combat tàctic. Originalment el combat havia de tindre lloc en una graella, on cada nau ocuparia un cert nombre de caselles depenent de la seva mida, es podrien moure una distancia que vindria determinada pel component motor de la nau i cada arma podria atacar des d'una distancia diferent.

Per desgràcia no vaig trobar una forma de fer la graella i que les naus s'adaptessin a ella. Així que al final vaig fer que el combat tingués lloc al mapa galàctic i que paràmetres de les naus com la velocitat o l'abast de les armes passessin a sumar punts al valor d'atac de la flota.

Classes

En aquest apartat es parlarà de les classes mes rellevants que formen part del joc. No surten totes les classes que es fan servir ja que, especialment les classes objecte, hi ha classes que son molt semblants les unes amb les altres.

Classes "MonoBehaviour"

Aquest tipus de classes hereten mètodes de MonoBehaviour i són més "interactuables" amb la resta del joc que les altres classes objecte.

Classe "GameController"

Com el seu nom indica, aquesta és la classe que controla tot el joc des de que es comença la partida fins que es torna al menú principal.

Els mètodes mes rellevants d'aquesta classe són els següents:

- *Awake*: Aquest mètode es crida automàticament quan es crea la primera instància d'un objecte que conté la classe *GameController* i fa dos coses: la primera és evitar que aquest es destrueixi al canviar d'escena, i la segona és evitar que hi hagin duplicats d'aquest objecte. D'aquesta manera tenim un únic controlador.
- *Start*: Aquest mètode es crida automàticament després d'executar el mètode *Awake*. Dins d'aquest mètode s'inicialitzen comptadors, s'afegeixen listeners³ als botons, es guarden els paràmetres de la galàxia que s'acaba de crear (sistemes solars, les seves posicions i quins jugadors hi ha) i es llegeixen els fitxers XML que corresponen a la llista d'edificis, la llista de components i la llista de tecnologies.
- *Update*: Aquest mètode es crida un cop cada frame i en ell es comprova en quina escena del joc es troba, i en cas de que es trobi en el mapa galàctic, comprova si s'ha fet clic en un sistema solar, i de ser així, obrirà la finestra amb les dades corresponents. Addicionalment també comprova les condicions de victòria de la partida.
- *UpdateUI*: Aquest mètode es crida cada cop que es fa clic en un sistema solar. Si la finestra amb les dades del sistema solar no existeix, se'n crea una de nova, i si ja n'hi ha una, se n'actualitzen les seves dades.
- *OpenBombing*: Aquest mètode es crida quan es fa clic sobre el botó de bombardejar un planeta. Aquí es comprova que les condicions per a bombardejar són correctes (el planeta pertany a un enemic i tens una flota que pugui atacar). Si són correctes, s'obre la finestra de bombardeig.

³ Un listener és una funció que es crida quan es compleixen certes condicions, com per exemple, fer clic en un botó.

- *NextTurn*: Aquest mètode es crida cada cop que es fa clic sobre el botó de següent torn. El que fa és recórrer tota la llista de sistemes solars i executar el mètode *ExecuteTurn* de cada un d'ells. A més a més fa avançar la investigació i quan es desbloqueja una nova tecnologia crea un pop-up per a avisar al jugador. Si hi ha una finestra amb dades d'un sistema solar oberta, al cridar aquest mètode s'actualitzaran les seves dades.
- Mètodes *HideStars* i *ShowStars*: Aquests dos mètodes es criden respectivament al començar i al acabar un bombardeig planetari. El mètode *HideStars* recorre la llista de sistemes solars, n'agafa el *GameObject* i el fa invisible, mentre que el mètode *ShowStars* fa l'inrevés. Aquests mètodes són necessaris ja que el bombardeig planetari té lloc a la mateixa escena que el mapa galàctic i hi havia interferències amb les imatges.

Classe "BombingController"

Aquesta classe és l'encarregada de fer controlar els bombardeigs planetaris. Els seus mètodes més rellevants son els següents:

- *Start*: Aquest mètode es crida automàticament després de crear l'objecte que conté aquesta classe. Aquí s'inicialitzen comptadors, s'assignen les imatges corresponents als sprites i es carrega l'àudio que es farà servir.
- *Update*: Aquest mètode es crida un cop cada frame i s'encarrega d'actualitzar les dades tant de la finestra, com les del planeta i la flota atacant. Addicionalment disposa d'un comptador per a tirar les bombes. D'aquesta manera s'evita que caiguin totes de cop i hi hagi interferències amb l'àudio.
- *BombSelector*: Aquest mètode es crida cada cop que s'apreta un dels vuit botons de bombardeig i fa que es tiri una bomba o que es posi a cert el booleà de bombardeig automàtic per a aquell tipus de bomba, depenent del botó en que s'hagi fet clic.
- *Bomb*: Aquest mètode es crida cada cop que es tira una bomba. Primer es comprova que es pugui bombardejar, i en cas afirmatiu crea una instància d'una explosió.
- *ActualizeCounters*: Aquest mètode és el que es fa servir dintre de *Bomb* per a comprovar si es pot bombardejar o no. Primer es mira amb quin tipus de bomba es vol atacar, i si n'hi ha una o més s'ataca i es decrementa el comptador d'aquell tipus de bombes i retorna cert. Si no es pot atacar retornarà fals i també posarà a fals el booleà de bombardeig automàtic per a aquell tipus de bomba.

Classe "TechManager"

Aquesta classe de la pantalla d'investigació és l'encarregada de seleccionar quines tecnologies es poden investigar d'entre totes les disponibles. Disposa dels mètodes següents:

- *Start*: Aquest mètode es crida automàticament després de crear l'objecte que conté aquesta classe. Aquí s'agafa l'arbre tecnològic de la classe *GameController*, juntament amb les tecnologies que ja s'han investigat.
- *SelectTech*: Aquest mètode es crida quan el jugador selecciona una nova tecnologia de l'arbre tecnològic, o bé es crida de forma externa quan s'ha acabat d'investigar una tecnologia. Quan es crida perquè el jugador ha escollit una nova tecnologia per a investigar, es fa des de la classe *TechViewer*⁴, i s'actualitza el text en pantalla que conté la informació pertinent a la tecnologia escollida i es crida el mètode *SetCurrentResearch* de la classe *GameController* per a que marqui aquesta tecnologia com la tecnologia a investigar.
- *SelectResearchableTech*: Aquest mètode es crida des del mètode *Start* i el que fa es crear una llista amb les tecnologies que es poden investigar. El requisit per a que una tecnologia és pugui investigar es que sigui la tecnologia més barata dintre d'aquella àrea d'investigació. Com que l'arbre tecnològic es guarda en un fitxer XML, si es volen afegir noves tecnologies no cal que aquestes estiguin ordenades ja que el propi joc les ordena per cost.
- *FillTechSelector*: Aquest mètode es crida des del mètode *Start* un cop ja tenim la llista de les tecnologies disponibles i el que fa és crear *GameObjects* amb els elements de la llista per a que el jugador pugui seleccionar-ne un.

⁴ La classe *TechViewer* es troba dins de cada un dels *GameObjects* de la llista de tecnologies amb la que l'usuari pot interactuar.

Classe "ShipyardManager"

Aquesta classe es troba en la pantalla de disseny de naus i s'encarrega d'afegir, treure i mostrar els components de la nau que s'està construint. Disposa dels mètodes següents:

- *Start*: Aquest mètode es crida automàticament després de crear l'objecte que conté aquesta classe. Aquí s'agafa la llista dels components disponibles de la classe *GameController*, s'inicialitzen els botons, s'afegeixen a la nau els components base i es creen les llistes de components ofensius i components especials.
- *Update*: Aquest mètode es crida un cop cada frame i s'encarrega d'actualitzar les dades dels paràmetres de la nau que surten en pantalla. També afegeix a la nau els components que s'hagin seleccionat abans de l'anterior crida a *Update*.
- *AddBaseComponents*: Aquest mètode es crida al mètode *Start* i s'encarrega de buscar a la llista de components disponibles tots els components base i escollir els més potents. Un cop els té els afegeix a la nau
- *AddWeapon* i *AddSystem*: Aquests mètodes es criden quan es fa clic sobre el botó d'afegir un component ofensiu o el botó d'afegir un sistema especial i s'encarreguen de crear una instància d'una nova finestra que conté una llista amb totes les armes o components especials disponibles.
- *AddWeaponOnUpdate* i *AddSystemOnUpdate*: Aquests mètodes es criden al fer clic sobre algun dels elements de les llistes creades als mètodes *AddWeapon* i *AddSystem*. El que fan és afegir l'arma o el sistema especial seleccionat i afegir-lo a la llista de components instal·lats pertinent i tancar les finestres amb les llistes de components.
- *RemoveWeapon* i *RemoveSystem*: Aquestes mètodes es criden al fer clic sobre el botó de treure el component seleccionat i s'encarreguen de treure el component seleccionat de la llista de components instal·lats i actualitzar la informació sobre la nau.
- *BuildDesign*: Aquest mètode es crida al fer clic sobre el botó de "Construir disseny" i si es compleixen els requisits per a construir una nau, procedeix a afegir-li tots els components seleccionats, passar a la pantalla de control de colònies i començar la seva construcció.

Classes objecte

Classe "Solar System"

Aquesta classe és l'encarregada d'emmagatzemar totes les dades pertanyents a un sistema solar: l'estrella, tots els seus planetes i qualsevol flota de naus que hi pugui haver.

Adicionalment, ja que es tracta d'un objecte interactuable, també té un `GameObject` i les seves coordenades per tal de recrear-lo un cop s'hagi canviat de pantalla.

Ignorant *getters* i *setters*, aquesta classe disposa dels mètodes següents:

- *Initialize()*: Aquest mètode es crida al constructor i assigna al `GameObject` el nom i el sprite de l'estrella del sistema solar. Aquest és el mètode que es crida quan es crea el sistema solar per primer cop.
- *Initialize(GameObject)*: Aquest mètode es crida des d'una classe externa cada cop que s'accedeix a la pantalla del mapa galàctic i s'han de recrear tots els sistemes solars.
- *GeneratePlanets()*: Aquest mètode es crida al constructor i genera entre un i cinc planetes aleatòriament i els afegeix a la llista de planetes del sistema solar.
- *IntToRoman(int)*: Aquest mètode es crida des de *GeneratePlanets()* i converteix un enter en un string que conté aquell enter en números romans.
- *ExecuteTurn()*: Aquest mètode es crida cada cop que es fa clic sobre el botó de "Següent torn". Un cop es crida executa, el mètode *ExecuteTurn()* de cada planeta, i si aquests han acabat de construir una nau, crea una nova flota amb ella. Un cop ha acabat, combinarà totes les flotes d'un mateix jugador en una sola flota.
- *ActualizeStarName()*: Aquest mètode es crida dins del mètode *ExecuteTurn()* i crea una llista amb els colors dels jugadors que controlen el sistema solar. En cas de que un sistema no el controli ningú, a la llista s'afegirà el color gris. Un cop té la llista crida al mètode *PaintName(List<Color>)*.
- *PaintName(List<Color>)*: Aquest mètode es crida al acabar *ActualizeStarName()* i rep per paràmetre una llista amb els colors que s'haurà de pintar el nom del sistema solar. El que fa és agafar el string del nom de l'estrella del sistema solar i partir-lo en tantes parts com colors diferents hi hagi a la llista. En acabar s'agafa cada una d'aquestes parts i se li dona un color fent servir el mateix format que en un text HTML, i després s'ajunten totes les parts de nou i es sobreescriu el text del `GameObject` que conté el sprite del sistema solar.

Classe "Planet"

Aquesta classe és l'encarregada d'emmagatzemar totes les dades pertanyents a un planeta: A qui pertany, quin clima té, quanta població hi ha, quina és la població màxima, quins edificis té construïts, etc.

Ignorant *getters*, *setters* i mètodes semblants, aquesta classe disposa dels mètodes següents:

- *GenerateBiome(string)*: Aquest mètode es crida des del constructor i el que fa és cridar al mètode *GenerateBiome()* que correspongui al tipus d'estrella que rep per paràmetre. Els altres mètodes *GenerateBiome()* el que faran és seleccionar aleatòriament un bioma pel planeta.
- *AssignPeople(int)* i *DismissPeople(int)*: Aquests mètodes es criden cada cop que s'apreta el botó d'assignar o el de desassignar una unitat de població en la pantalla de control de la colònia. El valor de l'enter determina en quin tipus de població es faran els canvis.
- *ExecuteTurn()*: Aquest mètode es crida des del mètode *ExecuteTurn()* de la classe "Solar System" i el que fa és comprovar si el planeta està controlat per un jugador, i de ser així executar els mètodes pertinents.
- *GenerateProduction ()*: Aquest mètode es crida des del mètode *ExecuteTurn()* i actualitza els valors de producció del planeta cridant els mètodes pertinents. Aquests mètodes el que faran és recórrer la llista dels edificis construïts, sumar els valors pertinents i multiplicar aquests valors pels bonus corresponents.

Exemple: Producció Industrial:

$$\text{Resultat} = \text{Població Industrial} * \text{Riquesa mineral}$$

De tots els edificis de tipus industrial que tinguin valors de producció directe:

$$\text{Resultat} = \text{Resultat} + \text{Valor de producció del edifici}$$

De tots els edificis de tipus industrial que tinguin valors de producció percentual:

$$\text{Multiplicador} = \text{Multiplicador} + \text{Valor de producció percentual}$$

Retorna:

$$\text{Resultat} + (\text{Resultat} * (\text{Multiplicador} / 100))$$

- *ManageBuildingConstruction()*: Aquest mètode es crida des del mètode *ExecuteTurn()* i s'encarrega d'agafar els valors de producció industrial i aplicar-los a l'objecte en producció. Si el valor de producció industrial és menor al valor requerit per a construir l'objecte, es guardaran per al pròxim torn.
- *UpdatePopulation()*: Aquest mètode es crida des del mètode *ExecuteTurn()* i s'encarrega d'afegir unitats de població fins que s'arribi al màxim.

- *ManageTroops()*: Aquest mètode es crida des del mètode *ExecuteTurn()* i el que fa és comprovar si hi han edificis que produeixin unitats per al combat terrestre, i de ser així, produir-ne una cada cop que el seu període d'espera arribi a zero.
- *ManagePollution()*: Aquest mètode es crida des del mètode *ExecuteTurn()* i comprova que el nivell de pol·lució es inferior al nivell màxim. Si es igual o superior, es canviarà el bioma del planeta a un bioma inferior i es tornarà a posar el nivell de pol·lució a zero.
Independentment de si es canvia el bioma o no, en acabar es cridarà un altre mètode que restarà al nivell de pol·lució el valor de la suma dels valors anti-pol·lució dels edificis que hi ha construïts.
- *MakeStartingPlanet(Jugador)*: Aquest mètode es crida al generar la galàxia i canvia les condicions del planeta per a que tots els jugadors comencin en igualtat de condicions.

Aspectes a afegir o millorar en futures versions del joc

- *Diplomàcia*: Un bon joc 4X té un sistema de diplomàcia que permet aliances, intercanvi de recursos o informació, i seria una bona alternativa al sistema que hi ha implementat ara mateix, en el qual tots els jugadors són enemics durant tota la partida.
- *Combat*: El combat en el joc ara mateix consisteix en sumar els valors de combat de la flota atacant i restar-los a la suma dels valors de combat de la flota defensiva. És un sistema que no té cap tipus d'estratègia i en el que la forma més fàcil de guanyar és fabricar tantes naus amb armes com sigui possible i esperar que l'enemic en tingui menys.
M'agradaria implementar un sistema com el del "*Masters of Orion 2*", tal i com volia fer originalment.
- *Música i so*: Actualment només hi ha so en les explosions del bombardeig planetari. Aniria bé tenir una sèrie de músiques de fons, així com també feedback al interactuar amb elements de la interfície d'usuari.
- *Millora gràfica*: La major part de les interfícies d'usuari són les genèriques d'Unity. Es dissenyar els gràfics d'una nova interfície. També es podria millorar el fons del menú principal i de la pantalla d'investigació.
- *Sistema de guardat i carregat de partides*: Pràcticament és un requisit per a fer un joc 4X, ja que les partides solen durar el suficient com per a no poder-les acabar en una única sessió de joc.
- *Multijugador*: Probablement es tracta del requeriment més difícil de tota la llista. Afegir un mode de joc per a més d'un jugador seria una millora més que acceptable per al projecte.

Annex

Tipus d'estrelles i les seves característiques

- Estrella blava

Són les estrelles més calentes i brillants. Tenen planetes molt secs, amb climes que varien entre l'àrid i el desèrtic fins al volcànic.



- Estrella vermella

Són els tipus d'estrelles més abundants. La seva temperatura i lluminositat poden variar, i per tant, els planetes que les orbiten poden tenir pràcticament qualsevol clima.



- Estrella groga

Un tipus d'estrelles amb una temperatura i lluminositat normals. Els planetes que les orbiten acostumen a tenir aigua en forma líquida i per tant, acostumen a ser molt fàcils d'habitar.



- Estrella blanca

De forma similar a les estrelles vermelles, les estrelles blanques acostumen a tenir planetes de qualsevol clima, tot i que tenen tendència a tenir planetes desèrtics.



- Gegant vermella

Una estrella massiva però de temperatura relativament baixa. La seva mida ha convertit els climes dels planetes que l'orbiten en inhòspits.



- Nana blanca

Un estel petit, vell, molt dens i de poca lluminositat que està al final de la seva vida. Acostumen a estar orbitats per planetes amb climes glacials o irradiats.



- Nana vermella

Les restes d'un antic estel, aquest tipus d'estrelles estan orbitats per planetes que havien tingut aigua en un passat, però que degut a les baixes temperatures ara estan coberts per una capa de gel.



- Protoestel

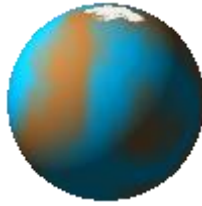
Una jove estrella que és poca cosa més que un núvol molecular molt dens. Es freqüent que els planetes que l'orbiten no siguin cap altra cosa més que planetoides de gran tamany.



Tipus de climes dels planetes

- Terraqui

Els planetes terraquis disposen de flora, fauna i aigua en abundància. Són els millors planetes per a viure-hi, però el seu ecosistema es delicat.



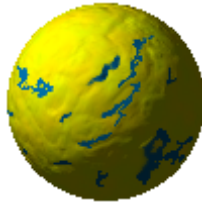
- Oceànic

Els planetes oceànics tenen la major part de la superfície coberta per grans masses d'aigua. La seva flora i fauna és susceptible a la pol·lució. Després dels planetes terraquis, són els millors planetes per a colonitzar.



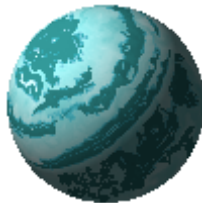
- Àrid

Els planetes àrids tenen un terreny rocós i poca quantitat d'aigua. La poca vegetació existent és bastant resistent a la pol·lució.



- Tundra

Els planetes amb aquest clima són freds i la vida a les seves glaceries és dura degut al temps inclement.



- Desèrtic

Els planetes desèrtics tenen condicions climàtiques adverses, altes temperatures i pràcticament ni una gota d'aigua en la seva superfície. Les elevades quantitats de sorra combinades amb tempestes de sorra freqüents fan que el manteniment dels edificis en aquests planetes sigui més car.



- Glacial

Els planetes glacials tenen tota la seva superfície coberta per una capa de diversos kilòmetres de neu i gel. Temperatures molt per sota dels zero graus i cap font d'aigua líquida dificulten la vida en aquest tipus de planetes.



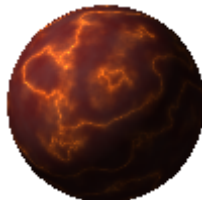
- Lunar

Aquests planetes presenten atmosferes molt tènues o pràcticament inexistentes que fan la vida a la superfície pràcticament impossible. La nul·la presència de flora i fauna fa que aquests planetes siguin molt tolerants a la pol·lució.



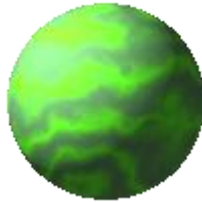
- Volcànic

Els planetes volcànics tenen una superfície extremadament calenta. L'alta activitat tectònica fa que els edificis necessitin reparacions mes sovint. Per sort l'abundància de matèria prima en aquests planetes pot mitigar els costs. La presència de volcans actius fa que desfer-se de productes contaminants sigui molt fàcil, de manera que els planetes volcànics resisteixen molt bé la pol·lució.



- Tòxic

Els planetes tòxics tenen climes extremadament hostils. Són planetes en els quals l'agricultura és completament impossible, i la població morirà lentament tret que hi hagin edificis que produeixin menjar. La pluja àcida i altres elements contaminants de l'atmosfera fan que el manteniment dels edificis sigui constant, incrementant així el seu cost. L'únic avantatge que tenen és que són immunes a la pol·lució: qualsevol contaminant que es generi no és res més que una gota en un oceà.



- Irradiat

Els planetes irradiats són planetes que han perdut la seva atmosfera degut a les emissions de la seva estrella. Són planetes en els quals l'agricultura és impossible, i la població es morirà lentament tret de que hi hagin edificis que produeixin menjar. L'alta radioactivitat a la que està sotmès el planeta fa que siguin immunes a la pol·lució.



Conclusions

En aquest projecte he implementat un joc d'estratègia 4X per torns inspirat en el "*Masters of Orion 2*". Tot i que les mecàniques i les pantalles del joc estan basades en aquest joc, he innovat en alguns aspectes, com per exemple en l'arbre tecnològic. Aquest es troba en un fitxer XML i això permet modificar un dels pilars del joc en un espai de temps extremadament curt. A més a més és tan fàcil de modificar que fins i tot un usuari sense experiència podria fer-hi canvis.

En les pròximes versions del joc tinc pensat millorar la intel·ligència artificial dels jugadors controlats per la màquina, ja que ara mateix els seus patrons de moviment són completament aleatoris i tant pot ser que t'ataquin directament tan aviat com puguin com que es passin tota la partida donant tombs pel mapa sense fer res. Addicionalment posaria una banda sonora i més efectes de so al joc ja que és una part que he deixat molt desatesa ja que al contrari que els sprites, interfícies i fons de pantalla no són indispensables per a poder jugar.

La realització d'aquest projecte no ha estat exempta de complicacions. Des de problemes lleus com que a l'hora de mostrar una escena els elements que formen part d'ella es trobessin a milers de píxels de distància de la pantalla a problemes més greus com per exemple que al fer la transició d'escenes entre la vista de la colònia i el mapa galàctic es dupliquessin planetes.

A banda d'aquests problemes que he pogut solucionar cal fer menció especial als múltiples problemes trobats a l'hora de fer el combat tàctic, principalment en el fet de crear una graella i aconseguir que els objectes actualitzessin les seves posicions prenent la graella com a referència.

En resum, fer aquest projecte no només m'ha ensenyat nous coneixements en Unity i m'ha permès millorar les meves habilitats amb l'eina GIMP, sinó que també m'ha ensenyat que fer un joc requereix molts coneixements en camps diferents i una gran quantitat de temps.

Bibliografia

- [1] Informació tècnica sobre "Masters of Orion 2": http://strategywiki.org/wiki/Master_of_Orion_II:_Battle_at_Antares
- [2] Sins of a Solar Empire: <https://www.sinsofasolarempire.com/>
- [3] Amplitude Studios, creadors de la saga Endless: <https://www.amplitude-studios.com/>
- [4] Civilization <https://civilization.com/>
- [5] Paradox, experts en jocs 4X <https://www.paradoxplaza.com/>
- [6] GIMP (GNU Image Manipulation Program) <https://www.gimp.org/>
- [7] Unity <https://unity3d.com/es>
- [8] Freesound.org <https://freesound.org/>

Totes les imatges i recursos gràfics del joc els he realitzat jo mitjançant l'eina GIMP.

Els sons del joc els he descarregat de freesound.org i pertanyen als seus respectius creadors.